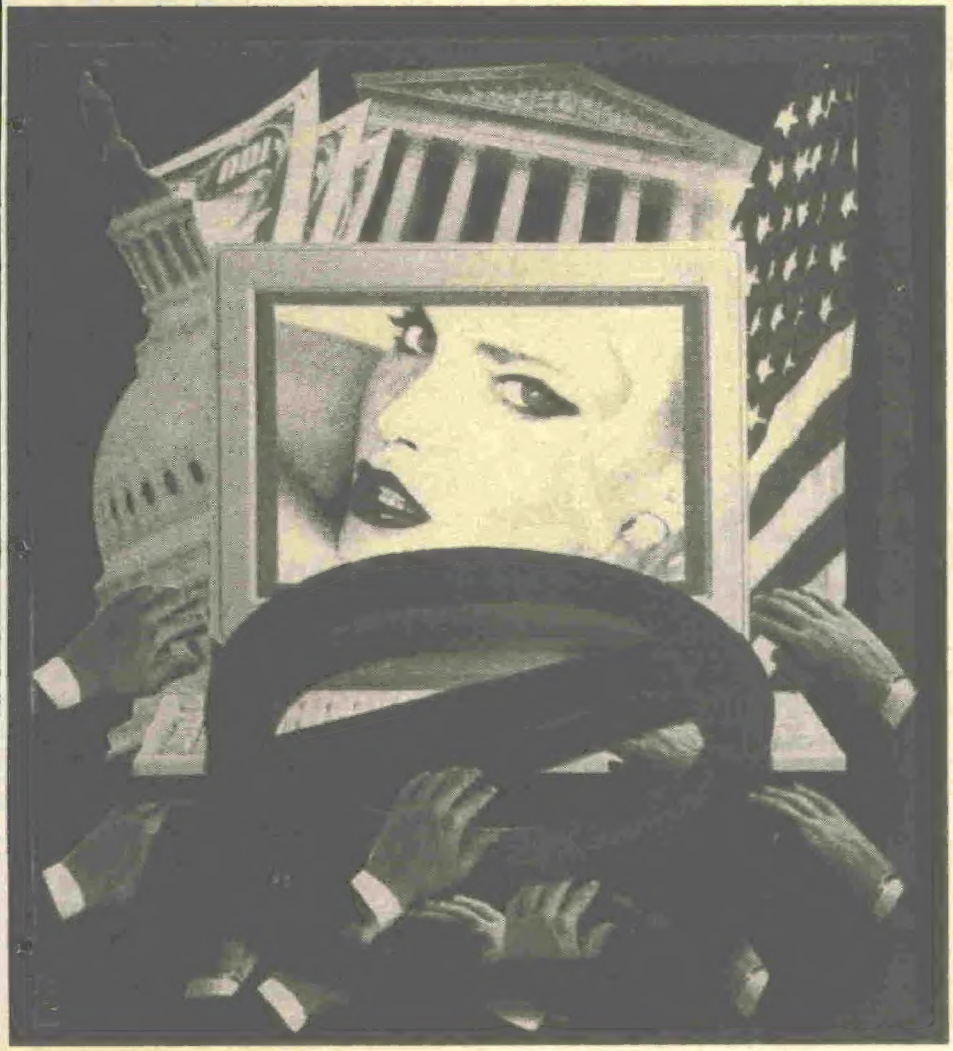
สัญญา เกษตรเวทิน

Delphi ผู้พิชิต

แนะนำขีดความสามารถของ Delphi สำหรับผู้ที่กำลังตัดสินใจวางแผนการพัฒนาโปรแกรมบนวินโดวส์



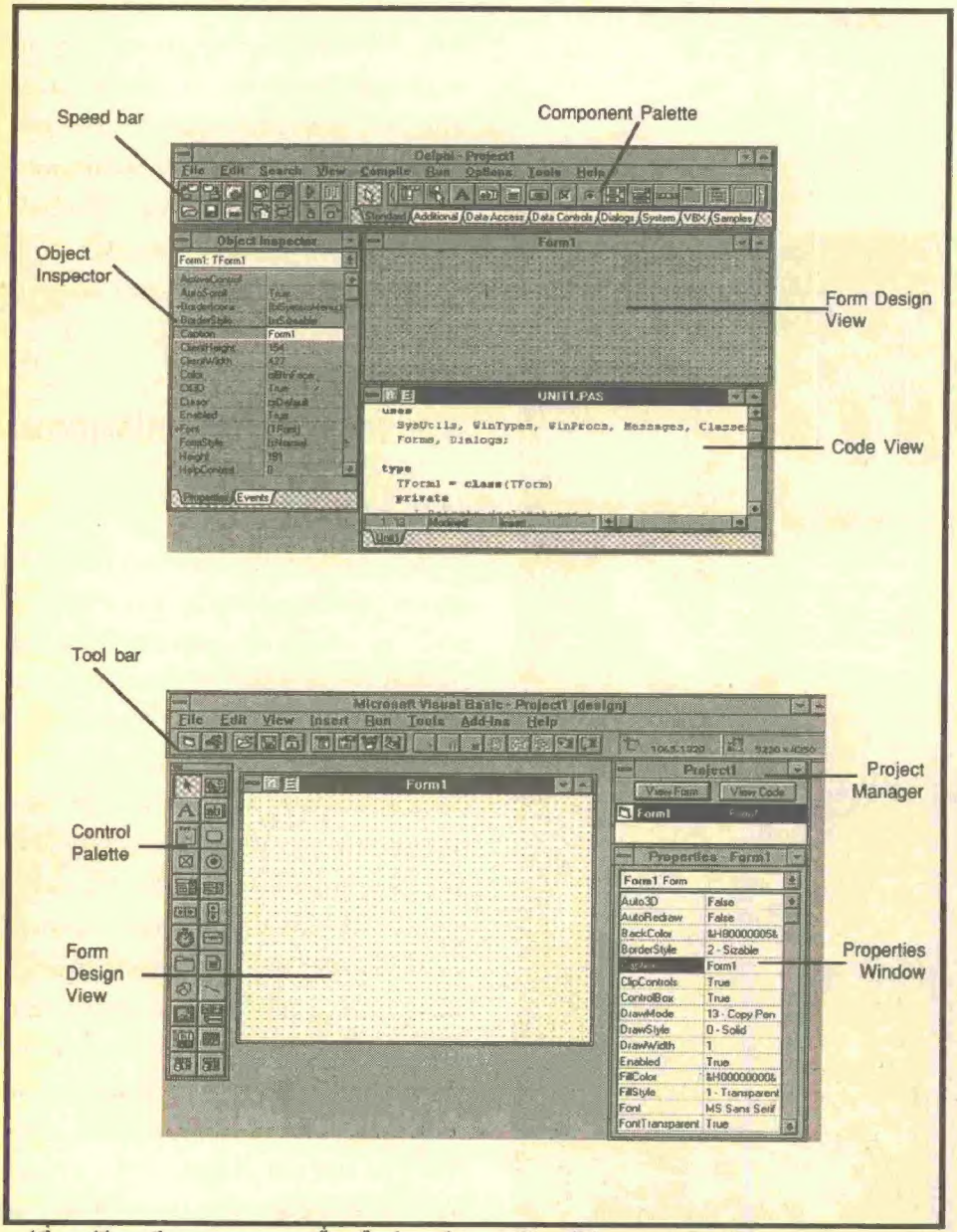
มองย้อนหลังไปเมื่อระบบปฏิบัติการของเครื่องคอมพิวเตอร์ส่วนบุคคลได้ถูกเปลี่ยนจากระบบของดอสมาเป็นวินโดวส์ การพัฒนาโปรแกรมให้ทำงานได้ภายในระบบก็ถูกเปลี่ยนตามไปด้วย หลายท่านที่เขียนโปรแกรมบนวินโดวส์มาก็คงจะทราบว่า การที่จะสร้างโปรแกรมสักโปรแกรมหนึ่งบนวินโดวส์นั้นแตกต่างจากระบบของดอสอยู่พอสมควร ด้วยความซับซ้อนแห่งระบบของวินโดวส์นี้เองบางครั้งทำให้การเขียนโปรแกรมกลายเป็นเรื่องยาก จนเป็นเหตุให้นักเขียนโปรแกรมมือสมัครเล่นบนดอสทั้งหลายต้องบอกลาวงการด้านการเขียนโปรแกรมไปเลยก็มี

แกะรอยการเขียนโปรแกรมแบบวิชวล

การพัฒนาโปรแกรมบนวินโดวส์ในยุคแรก ๆ จะเป็นการเขียนโปรแกรมด้วยภาษา C เป็นหลัก และมีการสร้างส่วนของโปรแกรมโครงสร้างพื้นฐานสำหรับการติดต่อกับส่วนต่าง ๆ ในวินโดวส์ ทำให้ดูเป็นการยุ่งยากเกินไป ประกอบกับบางคนที่ไม่คุ้นเคยกับภาษา C ก็จะเกิดความเบื่อหน่ายไปได้ง่าย ๆ ก้าวต่อมาในยุคของการเขียนโปรแกรมแบบ OOP ก็ได้มีการนำเอาภาษาอื่น ๆ มาประยุกต์ใช้งานและปรับปรุงให้เหมาะสมกับการทำงานในสภาพแวดล้อมของวินโดวส์ ส่วนติดต่อกับวินโดวส์ก็มีการปรับปรุง โดยเก็บอยู่ในส่วนของไลบรารี ทำให้การเขียนโปรแกรมมีขนาดกะทัดรัดลง แต่ถึงอย่างไรก็ตาม โครงสร้างในการควบคุมส่วนใหญ่ผู้ใช้ยังคงต้องเขียนสั่งงานเองแทบทั้งสิ้น แม้กระทั่งการออกแบบส่วนติดต่อ (Interface) ทำให้ต้องใช้เวลาในการพัฒนาโปรแกรมกันค่อนข้างนานกว่าจะได้มาสักโปรแกรมหนึ่ง

เมื่อมาถึงยุคที่สามแห่งการพัฒนาโปรแกรมบนวินโดวส์ คือยุคแห่งภาษาวิชวล หรืออาจจะเรียกว่าการโปรแกรมด้วยภาพ ทำให้การเขียนโปรแกรมบนวินโดวส์ดูจะไม่ใช่เรื่องที่ยากเย็นอะไรอีกต่อไป (อาจจะเรียกได้ว่าง่ายกว่าบนดอสหลายเท่า) ผู้ใช้เพียงแต่วางออปเจ็กต์ลงบนฟอร์มเพื่อกำหนดส่วนติดต่อหรือรูปแบบในการใช้งาน แล้วกำหนดหน้าที่ที่ต้องการใช้มันกระทำ ก็เป็นอันเสร็จ

ภาษาที่นับได้ว่าเป็นภาษาแรกที่ได้มีการพัฒนากันให้ใช้งานได้อย่างค่อนข้างจะจริงจังก็เห็นจะเป็น Microsoft Visual BASIC สำหรับวินโดวส์ ซึ่งก็ได้ใช้ทั้งโครงสร้างและรูปแบบของภาษา BASIC มาเป็นแนวทางในการเขียน ทำให้ผู้ที่ไม่รู้ภาษาคอมพิวเตอร์ก็สามารถที่จะศึกษาการเขียนโปรแกรมได้ภายในเวลาเพียงไม่กี่วัน (แต่ถ้าจะเขียนแบบมืออาชีพก็ต้องฝึกกันนานหน่อย) ด้วยความง่ายของภาษาทำให้ Visual BASIC (ขอเรียกย่อ ๆ ว่า VB) ได้รับการพัฒนามาจนถึงเวอร์ชันที่ 4.0 และสามารถทำงานบนระบบปฏิบัติการแบบ 32 บิตอย่าง Windows 95 ได้อีกด้วย และด้วยประสิทธิภาพที่ยอมรับได้ ผู้พัฒนาโปรแกรมหลายคนก็เปลี่ยนมาพัฒนาโปรแกรมบน VB กันเป็นส่วนมาก เนื่องจากงานบางอย่างสามารถที่จะลดต้นทุนลงไปได้เมื่อเทียบกับการพัฒนาโปรแกรมด้วยภาษา C/C++, Pascal หรือภาษาอื่น



รูปที่ 1 เปรียบเทียบสภาพของเครื่องมือต่าง ๆ ในการพัฒนาโปรแกรมของ Delphi และ VB

และเมื่อประมาณเดือนมกราคม ปี 2538 บริษัท Borland ซึ่งเป็นยักษ์ใหญ่แห่งวงการซอฟต์แวร์เจ้าหนึ่ง ก็ได้วางจำหน่ายเครื่องมือสำหรับการพัฒนาโปรแกรมแบบวิชวลบนวินโดวส์ตัวใหม่ ชื่อว่า Delphi อย่างเป็นทางการ (อ่าน "เดลฟี" หรือ "เดลไฟ" ก็แล้วแต่จะสะดวก) ภายหลังจากที่ได้มีการเปิดตัวกันมา 3-4 เดือนก่อนหน้านี้ (เมืองไทยเพิ่งจะมีการนำเข้าอย่างเป็นทางการเมื่อไม่กี่เดือนมานี้) จนได้รับเสียงวิพากษ์วิจารณ์กันว่ามันจะเป็นภาษาใหม่ที่เข้ามาแทนที่ VB ได้อย่างสบาย ด้วยประสิทธิภาพในการทำงาน ความง่ายในการเขียนโปรแกรมบนวินโดวส์ และความเร็วในการทำงานที่มากกว่าภาษาอย่าง VB หลายเท่าตัว

Delphi หรืออาจจะเรียกว่า Visual Pascal ก็คงจะไม่แปลกอะไรนัก ด้วยลักษณะของการพัฒนาโปรแกรมที่เป็นแบบวิชวลเหมือนกับ VB แต่การทำงานภายในจะแตกต่างกันบ้างเนื่องจากโครงสร้างของภาษาเดิม เรามาลองสำรวจดูขีดความสามารถของโปรแกรม Delphi ดูบ้างว่าจะสามารถมาแทนที่ VB ได้หรือไม่

บนเส้นทางของการพัฒนาโปรแกรม

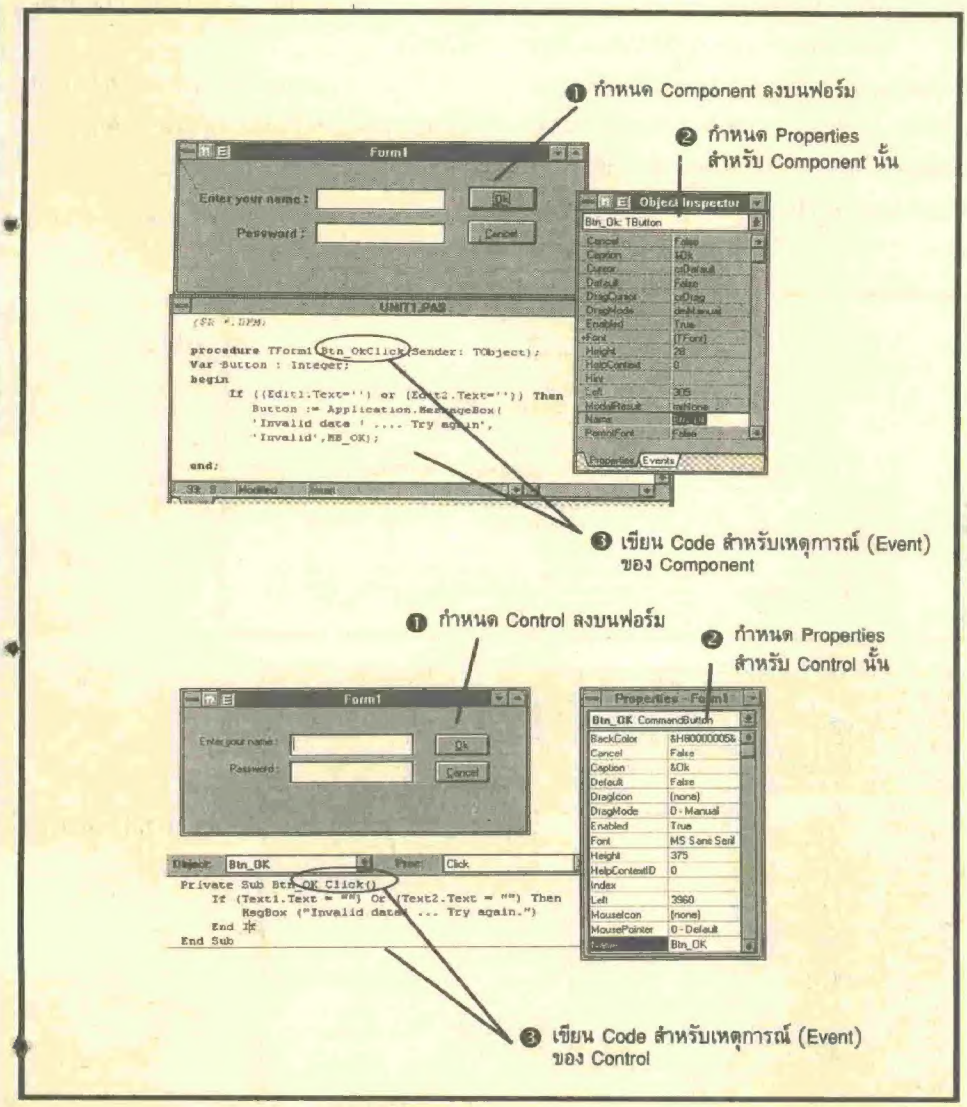
ทั้ง Delphi และ VB ก็เป็นภาษาในลักษณะของ Visual Programming ด้วยกันทั้งคู่ ดังนั้นเมื่อเราเปรียบเทียบดูการออกแบบเครื่องมือในการพัฒนาโปรแกรมจากรูปที่ 1

ก็เรียกได้ว่าคล้ายกันมาก แต่ชื่อเรียกอาจจะแตกต่างกันบ้างเท่านั้น ยกตัวอย่างเช่น Object Inspector (Delphi) และ Properties (VB), Component (Delphi) และ Control (VB) เป็นต้น นอกนั้นก็จะดูคล้าย ๆ กัน ทำให้ผู้ที่คุ้นเคยอยู่กับ VB สามารถเปลี่ยนมาใช้ Delphi ได้โดยไม่ยากนัก

สำหรับจุดที่แตกต่างกันอย่างเห็นได้ชัดก็คือรูปแบบของภาษา ซึ่ง Delphi ใช้ภาษา Object Pascal (คล้าย Turbo Pascal 7.0) ส่วน VB ใช้ภาษาที่พัฒนามาจาก Quick BASIC บนดอส

แนวทางในการเขียนโปรแกรม

รูปแบบและโครงสร้างของโปรแกรมเป็นการทำงานในลักษณะของ Event Driven Programming โดยการวางคอนโทรลหรือคอมโพเนนต์ต่าง ๆ ลงบนฟอร์ม กำหนด Property และกำหนดส่วนของ Code (หากจำเป็น) โดยเครื่องจะเป็นผู้กำหนดส่วนของโปรแกรมหลัก ๆ ไว้ให้เรียบร้อยแล้ว เราก็เพียงแต่เติมส่วนของโปรแกรมบางส่วนสำหรับออปเจ็กต์ที่กำหนดเข้าไปใหม่เท่านั้น (ดูรูปที่ 2 ประกอบ)



รูปที่ 2 แสดงลักษณะของการพัฒนาโปรแกรมแบบวิชวลของ Delphi และ VB

แนวทางของบทความนี้

ในบทความนี้จะแสดงให้เห็นถึงลักษณะโดยทั่วไปและขีดความสามารถของโปรแกรม Delphi เมื่อเทียบกับโปรแกรม Visual BASIC เพื่อเป็นแนวทางให้ผู้อ่านพิจารณาและเป็นการแนะแนวสำหรับผู้ที่กำลังคิดว่าจะเปลี่ยนมาใช้ Delphi โดยผู้เขียนจะไม่กล่าวถึงรายละเอียดในตัวโปรแกรมมากนัก เนื่องจากได้มีบทความหลายบทความกล่าวถึงกันบ้างแล้ว

ขีดความสามารถของ Delphi

ขีดความสามารถของโปรแกรม Delphi ที่ทางบอร์แลนด์ได้โฆษณามาโดยมากจะกล่าวถึง

- การพัฒนาโปรแกรมที่มีขีดความสามารถในการทำงานได้เป็นอย่างดีบนวินโดวส์ รวมถึงเรื่องของความเร็วในการทำงาน ความคล่องตัวของโปรแกรม (ประกอบด้วยไฟล์จำนวนน้อย)

- ความสามารถในการนำเอาโค้ดตลอดจนออปเจ็กต์ต่าง ๆ มาใช้ใหม่ได้ หรือนำไปใช้กับโปรแกรมอื่นได้อีกโดยไม่ได้เขียนใหม่ และสามารถที่จะปรับเข้ากับระบบปฏิบัติการแบบ 32 บิตได้ทันทีโดยไม่ต้องแก้ไขส่วนของโค้ด

- ความสะดวกและรวดเร็วในการพัฒนาโปรแกรม ด้วยระบบการพัฒนาโปรแกรมแบบวิชวล

- การใช้งานระบบฐานข้อมูลที่อ่อนตัวสามารถเชื่อมต่อกับระบบใหญ่ได้ทันที

จากความสามารถนี้เรามามองในรายละเอียดเป็นส่วน ๆ ดูกันบ้าง ดังนี้

รวดเร็วในการแปลภาษา

ความสามารถในการแปลส่วนของโค้ด สามารถแปลได้มากกว่า 350,000 บรรทัดต่อนาที (วัดบนเครื่อง Pentium) ซึ่งหากเทียบดูก็คงจะเป็นตัวแปลภาษาที่เร็วที่สุดในโลกก็ว่าได้ โดยจะทำการแปลเป็นคำสั่งที่ทำงานได้ทันทีโดยไม่ต้องนำไปแปลใหม่อีกครั้งก่อนการทำงาน จึงทำให้ทำงานได้เร็วกว่าภาษาที่เป็นแบบ Interpreter 10-20 เท่า

โปรแกรมที่ทำการแปลเรียบร้อยแล้วสามารถนำไปใช้ได้ทันที ไม่ต้องมีการเรียกใช้โปรแกรมประเภท DLL เพิ่มเติม (หรือจะเขียนให้เรียกใช้ก็ได้) และโปรแกรมที่ได้ก็นำไปใช้ได้อย่างไม่มีข้อจำกัดใด ๆ

นอกเหนือจากการพัฒนาโปรแกรมใช้งานหลัก ๆ แล้ว Delphi ยังสามารถที่จะสร้างไฟล์ประเภท DLL (Dynamic Link Library) และคอมโปเนนต์ (Component) เช่นเดียวกับไฟล์ VBX ได้อีกด้วย ซึ่งแต่เดิมเราจะต้องสร้างด้วยภาษา C หรือ C++

ความสะดวกและรวดเร็วในการพัฒนาโปรแกรม

การเขียนโปรแกรมรวมถึงการออกแบบส่วนในการติดต่อกับผู้ใช้กระทำในลักษณะของวิชวล ซึ่งทำให้เราสามารถสร้างโปรแกรมได้สะดวก เพียงแต่เลือกคอมโปเนนต์ (หรือเรียกว่าคอนโทรล - Control ใน VB) ที่ต้องการมาจาก Palette ซึ่งได้เตรียมมาไว้ใช้มากกว่า 75 คอมโปเนนต์ (นับได้ 78 คอมโปเนนต์) รวมถึงคอมโปเนนต์ด้านกราฟิกมัลติมีเดีย การควบคุมมาตรฐานต่าง ๆ การควบคุมฐานข้อมูล ไดอะล็อก เป็นต้น และนอกจากนี้ยังสามารถที่จะนำเอาไฟล์ VBX ของ Visual BASIC มาใช้ได้อีกด้วย

ความสามารถด้านการจัดการฐานข้อมูล

"No matter where your data resides, Delphi can reach it." การติดต่อกับไฟล์ฐานข้อมูลไม่ว่าจะเป็นระบบเล็กหรือระบบใหญ่ Delphi จะติดต่อผ่านทาง Borland Database Engine (BDE) และเชื่อมต่อด้วยภาษา SQL มาตรฐาน ทำให้สามารถติดต่อกับโปรแกรม Oracle, Sybase, Informix, Microsoft SQL Server และ Interbase ได้อย่างไม่มีปัญหา รวมถึงการติดต่อกับฐานข้อมูลผ่านทาง ODBC ด้วย (ดูรูปที่ 3 ประกอบ)

ในระบบของ Delphi ก็ยังมีโปรแกรม Local Interbase Server ติดมาให้ด้วย สำหรับการพัฒนาโปรแกรมด้วย SQL ในตัวเป็นการเตรียมพร้อมสำหรับการติดต่อกับโปรแกรมฐานข้อมูลขนาดใหญ่ภายนอก

เรียน Delphi กับ Interactive Tutor

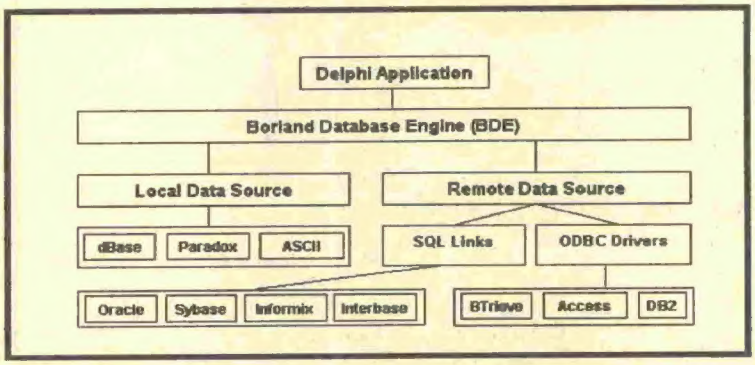
ในโปรแกรมของ Delphi เมื่อทำการติดตั้งแล้วผู้ใช้สามารถศึกษาการใช้งานได้อย่างง่าย เพราะในชุดโปรแกรมของ Delphi จะประกอบด้วย

1. On-line Interactive Tutor - โปรแกรมแนะนำเบื้องต้น

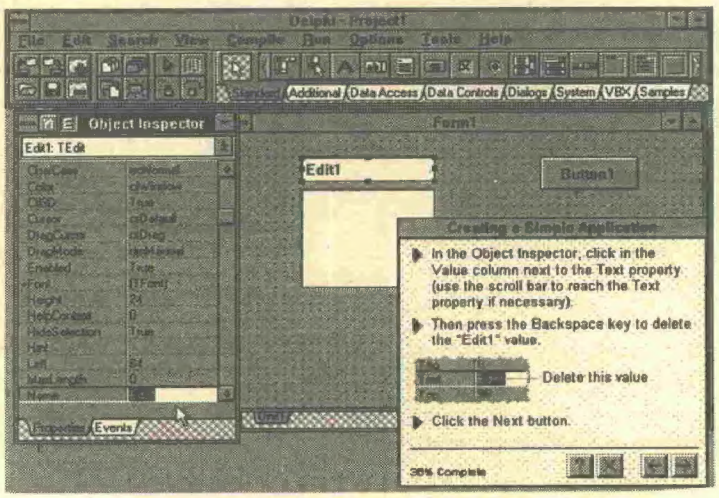
2. Complete On-line Document - คู่มืออิเล็กทรอนิกส์ (Help File)

3. Experts and Application Templates - เครื่องมือช่วยในการสร้างแอพพลิเคชัน

4. Example Project - ตัวอย่างโปรแกรมเพื่อใช้เป็นแนวทางในการศึกษา



รูปที่ 3



รูปที่ 4
แสดง Interactive Tutors

หากเราเริ่มใช้งาน Delphi ใหม่ ๆ ก็ไม่ควรที่จะพลาดการศึกษาลักษณะทั่ว ๆ ไปของ Delphi จาก On-line Tutor ในส่วนนี้ก่อน โดยจะอยู่ในเมนู Help->Interactive Tutors หน้าตาก็จะเป็นดังรูปที่ 4

หัวข้อใน Tutors จะประกอบไปด้วย

1. Introducing Delphi
1.1 A Quick look at Delphi
1.2 Creating a simple application
2. Creating User Interface
2.1 Adding Components to a form
2.2 Setting component properties
2.3 Writing Event Handlers
3. Database Applications
3.1 Setting up a table view
3.2 Querying a table

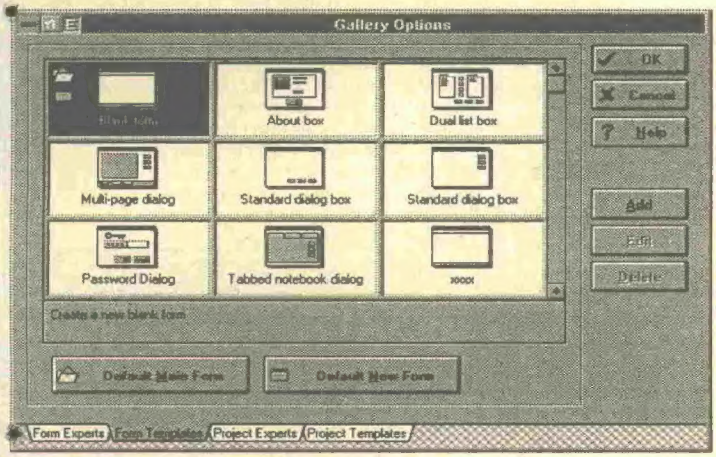
ทุกหัวข้อใน Tutor ผู้ใช้จะต้องทำตามไปด้วย ซึ่งจะช่วยให้เข้าใจและเห็นแนวทางการพัฒนาโปรแกรมด้วย Delphi พอสมควร โดยเฉพาะในเรื่องของฐานข้อมูล หากเปรียบเทียบกับโปรแกรมอื่น ๆ แล้วก็นับว่ามีความสะดวกในการออกแบบและใช้งานมาก อาจเรียกได้ว่า สร้างโปรแกรมติดต่อกับฐานข้อมูลได้เพียงปลายนิ้ว

สร้างฟอร์มจาก Template

เมื่อเราดูลักษณะของฟอร์มที่ใช้งานบนวินโดวส์โดยทั่วไปก็จะมีอยู่ไม่กี่แบบ อาจจะเรียกได้ว่าเป็นแบบมาตรฐานก็ได้ Delphi ได้กำหนดแนวทางการออกแบบโปรแกรมในรูปแบบนี้ไว้เป็นมาตรฐานแล้ว โดยบรรจุไว้ใน Templates และจะกำหนดการเรียกใช้ได้โดยต้องไปกำหนดค่าจากเมนู Options->Environments->Preferences->Gallery โดยกำหนดให้เลือกจากแบบ Template ที่มีอยู่แล้ว (ดังรูปที่ 5) หลังจากนั้นเมื่อเราเลือกหัวข้อ New Project (หรือ New Form หากต้องการสร้างเฉพาะในส่วนของฟอร์ม) ก็จะสามารถสร้างโปรแกรมในระดับมาตรฐานได้อย่างง่ายดาย

นอกจากนั้นเราสามารถที่จะเก็บโปรแกรมที่เราสร้างไว้แล้วไว้เป็น Template ได้ด้วยเหมือนกัน วิธีการที่จะเก็บโปรแกรมไว้เป็น Template ก็เพียงแต่คลิ๊กปุ่มขวาของเม้าส์ไปที่ฟอร์มที่ออกแบบ และเลือกรายการ Save as Template... กำหนดหัวข้อและรายละเอียดที่ต้องการ หรือจะเลือกจากรายการในหัวข้อ Options=>Gallery... ก็ได้เช่นกัน

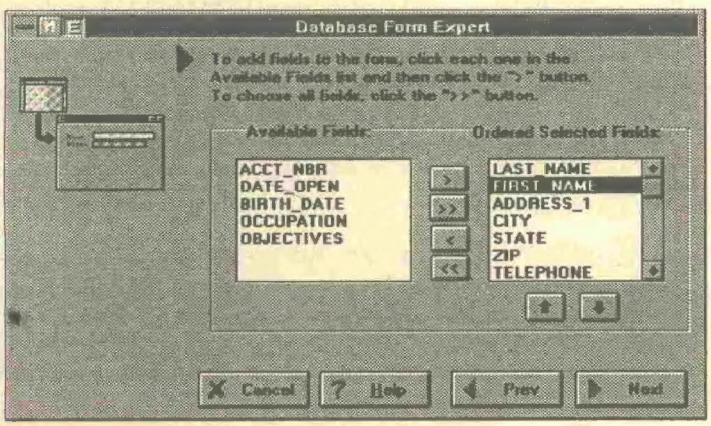
หากไม่พอใจกับสิ่งที่เครื่องสร้างมาให้สำเร็จแล้วทั้งหมดก็สามารถสร้างโปรแกรมในลักษณะคล้ายกับ Wizards ในรูปแบบของ Microsoft ได้อีกด้วย โดยเลือกจาก Expert Gallery



รูปที่ 5
แสดงการสร้างฟอร์มจาก Template

ฐานข้อมูลจาก Database form expert

ในเมนูย่อยของ Help จะมีรายการให้เลือกสำหรับการสร้างโปรแกรมทางด้าน Database โดยเครื่องจะเป็นผู้กำหนดแนวทางในการออกแบบให้และให้ผู้ใช้เลือกในหัวข้อที่ต้องการ หรือใส่ข้อมูลที่เครื่องควรจะทราบก่อนการสร้างโปรแกรม โดยเลือกจากรายการ Help->Database form expert... หลังจากนั้นก็ตอบคำถามตามที่เครื่องถามมา คลิ๊กที่ปุ่ม Next เพื่อทำต่อไป หรือคลิ๊กปุ่ม Create เพื่อสร้างโปรแกรมจัดการฐานข้อมูล เพียงเท่านี้ท่านก็สามารถสร้างโปรแกรมจัดการฐานข้อมูลด้วยตัวท่านเองแล้ว



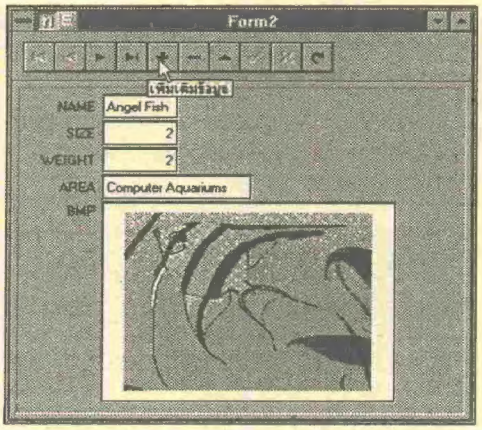
รูปที่ 6
แสดง Database form Expert คล้าย Wizard ของไมโครซอฟต์

Visual Component Library (VCL)

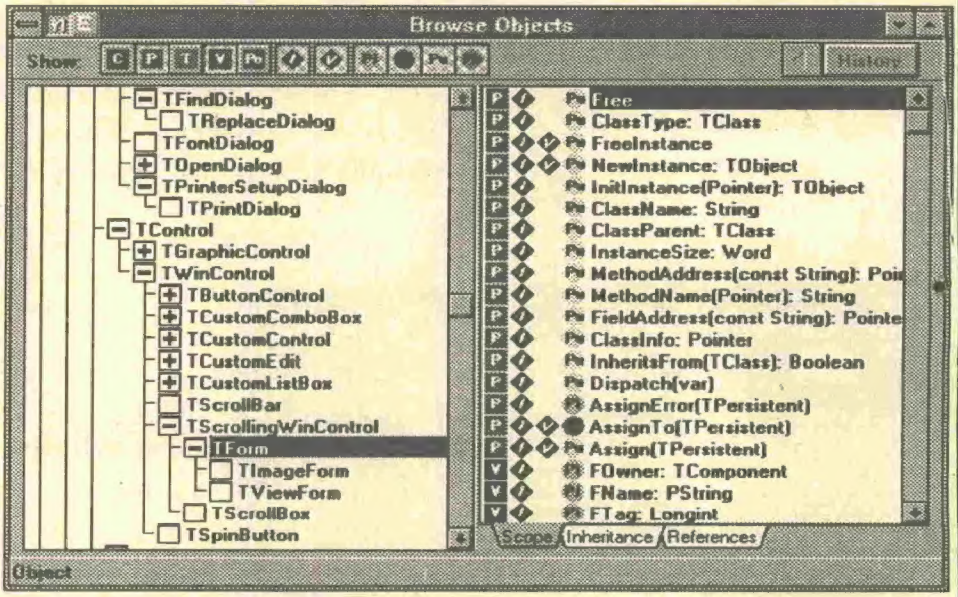
ส่วนประกอบต่าง ๆ ที่กำหนดไว้ในฟอร์ม หรือออปเจ็กต์ใดที่สร้างไว้ในโปรแกรม เราเรียกว่า Component ซึ่งจะถูกเก็บรวมไว้เป็นไลบรารีไฟล์ ซึ่งทั้งหมดจะเป็นไฟล์ที่ถูกสร้างขึ้นด้วย Delphi เอง โดยใช้รูปแบบเดียวกับที่กำหนดโครงสร้างให้กับผู้ใช้ ซึ่งจะทำให้ผู้ใช้สามารถพัฒนาโปรแกรมภายใต้สภาพของ Delphi ได้อย่างสมบูรณ์

โดยทั่วไป VCL จะประกอบด้วยออปเจ็กต์หรือคอมโปเนนต์ต่าง ๆ ซึ่งเรามักจะเรียกติดปากกันว่า "Controls" (ดังที่เรียกกันใน Visual BASIC) ประกอบด้วยส่วนที่เป็นออปเจ็กต์มาตรฐาน เช่น Label, Combo Box, List Box เป็นต้น และส่วนที่ได้กำหนดเพิ่มเติมเข้ามา เช่น Tab, Notebook เป็นต้น ซึ่งออปเจ็กต์เหล่านี้ก็ได้ถูกบรรจุไว้เป็นไลบรารีที่ผู้ใช้สามารถเรียกใช้ได้ทันที เมื่อทำการคอมพายล์ก็จะนำเอาไลบรารีส่วนนี้ติดไปด้วย ทำให้ไม่ต้องเรียกใช้ข้อมูลจากไฟล์อื่น

Component ทุกตัวสามารถที่จะกำหนด Hint หรือคำอธิบายย่อยในโปรแกรมได้อีกด้วย (บางคนก็เรียก Tool tips หรือ Help Ballon ตามแบบฉบับของเครื่อง Macintosh) ใน Hint นี้ผู้ใช้สามารถกำหนดได้เอง ดูรูปที่ 7 ประกอบ



รูปที่ 7
แสดงการกำหนด Hint และแสดงขณะที่โปรแกรมทำงาน



รูปที่ 8 แสดง Object Browser จากไฟล์ ImageView

ในส่วนของออปเจ็กต์เหล่านี้ถูกสร้างขึ้นมาเรียบร้อยแล้ว โดยกำหนดเป็นลักษณะของ Unit ใน Pascal แล้วทำการคอมพายล์เก็บไว้เป็นส่วนของไลบรารีเอง หากผู้ใช้ต้องการที่จะเขียนขึ้นมาใหม่ก็สามารถศึกษาแนวทางการเขียนจากตัวอย่างโปรแกรมที่มีให้แล้ว ลองศึกษาดูจากไดเรคทอรี \Delphi\Source\Samples ซึ่งมีตัวอย่างในการเขียนออปเจ็กต์ของปฏิทิน รวมถึงการควบคุมในส่วนของพรอพเพอร์ตี้ส์ต่าง ๆ อีกด้วย

อีกลักษณะหนึ่งซึ่งก็เป็นการท้าทายสำหรับผู้ที่เข้าใจในหลักการของภาษาแบบ OOP ก็คือ การสร้างออปเจ็กต์ใหม่จากการสืบทอด (Inheritance) มาจากออปเจ็กต์เดิม ซึ่งบอร์แลนด์ก็มีตัวอย่างพร้อมคำอธิบายมาให้เรียบร้อย ลองอ่านดูได้จากไฟล์ INSTALL.TXT

นอกจากนั้นไฟล์จำนวนมากที่มีนามสกุลเป็น VBX ซึ่งถูกเรียกใช้โดยโปรแกรมบน Visual BASIC ก็สามารถที่จะนำเข้ามาใช้ใน Delphi ได้เหมือนกัน ดูแล้วบอร์แลนด์ก็เข้าใจทำเหมือนกัน ทำให้เราไม่ต้องไปเสียดายกับไฟล์ VBX ที่มีอยู่จำนวนมาก แต่เมื่อเรามองดูอีกทีก็จะพบว่า บรรดาไฟล์ VBX ที่นิยมใช้กันส่วนมากนั้นก็ได้ถูกบรรจุไว้ในไลบรารีไฟล์เรียบร้อยหมดแล้ว แทบจะเรียกว่าไม่ต้องใช้ไฟล์ VBX เลยก็สามารถสร้างโปรแกรมที่มีรูปร่างหน้าตาระดับมืออาชีพได้อย่างสบาย

Object Browser

เป็นเครื่องมือช่วยที่ช่วยให้ศึกษาโครงสร้างของออปเจ็กต์ที่สร้างขึ้นในโปรแกรมได้เป็นอย่างดี เมื่อเราคอมพายล์โปรแกรมเรียบร้อยแล้ว ลองเลือก View => Browser จากเมนู เราจะได้วินโดว์ของ Object Browser ดังแสดงในรูปที่ 8 ปุ่มขวาของเม้าส์ทำงานได้ดีมาก เมื่อเราคลิ๊กก็จะเกิดเมนูย่อยขึ้นมาให้เลือก หากเรา

อยากจะเลือกดู Unit ที่ได้คอมพายล์ไว้แล้ว และแสดงถึงส่วนที่คุณสามารถจะเรียกใช้งานได้อีกด้วย ในวินโดว์ของ Object Browser จะแบ่งเป็น 2 ส่วน โดยส่วนทางขวาจะแสดงถึงรายละเอียดเพิ่มเติมจากที่เราได้เลือกในทางซ้าย ดังตัวอย่างจากรูป เราจะเห็นออปเจ็กต์ของ TImageForm สืบทอดมาจากออปเจ็กต์ Tform และออปเจ็กต์ Tform นี้ก็ได้สืบทอดมาจาก TscrollingWindowControl และ TWinControl อีกครั้งหนึ่ง และสืบทอดกันไปเรื่อย ๆ จนถึง Tobject ซึ่งถือว่าเป็น Parent Object ของคลาสทั้งหมด (ใน Delphi จะใช้ตัวอักษร T นำหน้าเพื่อแสดงว่าเป็นชนิดข้อมูลของออปเจ็กต์) ส่วนในช่องทางขวาเราก็จะเห็นตัวแปรและโพรซีเดอร์ต่าง ๆ ที่กำหนดไว้ในออปเจ็กต์ของ TImageForm

ปุ่มบน Speedbar ด้านบนใช้สำหรับกำหนดฟิลเตอร์ต่าง ๆ เพื่อควบคุมการแสดงผลเฉพาะรายละเอียดที่ต้องการ โดยสัญลักษณ์จากซ้ายไปขวากำหนดไว้ดังนี้

- Constants, Procedures, Types
- Variables, Properties
- Inherited, Virtual methods
- Private, Protected, Public, Published fields

ปุ่มต่าง ๆ จะเป็นลักษณะ Toggle ทำให้เราสามารถกำหนดการแสดงผลเฉพาะส่วนต่าง ๆ ได้สะดวก ส่วนในช่องทางด้านซ้ายก็จะเป็นลักษณะของ Tree ที่เราสามารถขยายหรือหดได้ด้วยการคลิ๊กที่เครื่องหมาย "+" หรือ "-" คล้าย ๆ กับการควบคุมใน File Manager

ภาษา Object Pascal

ภาษาที่ใช้ในการเขียนโปรแกรมบน Delphi ใช้ภาษา Pascal ของบอร์แลนด์ ที่เขียนในลักษณะของ Object-Oriented หรือที่เรียกกันสั้น ๆ ว่า Object Pascal ซึ่งมีความแตกต่างจากภาษา Pascal มาตรฐาน ในเรื่องของประสิทธิภาพและขีดความสามารถในการทำงาน แต่ก็ยังคงรูปแบบในการที่เป็นภาษาโครงสร้างอยู่เหมือนเดิม ทำให้การเขียนโปรแกรมที่มีขนาดใหญ่ลดความซับซ้อนลงไปได้มาก และด้วยลักษณะของการใช้คำในภาษาเองทำให้ผู้ใช้สามารถเข้าใจในโครงสร้างและตัวโปรแกรมได้ง่าย ลดเวลาในการศึกษาในตัวภาษาลงได้เป็นอันมาก

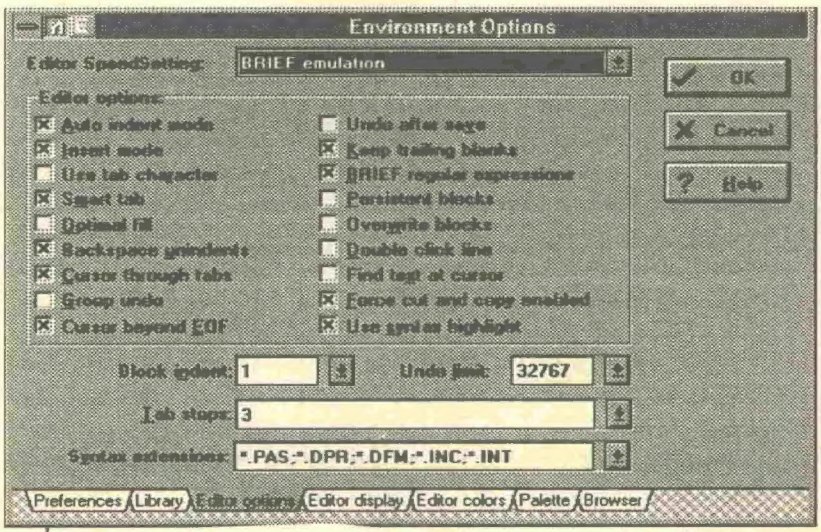
Object Pascal ได้ถูกนำมาใช้เป็นภาษาที่สนับสนุนการทำงานภายใน Delphi ด้วยการเขียนโปรแกรมในลักษณะเดียวกับระบบของคอส ทำให้ผู้ใช้สามารถพัฒนาโปรแกรมถึงระดับ Client/Server บนวินโดวส์ได้โดยที่ไม่ต้องมีความรู้ในเรื่องของ messages ต่าง ๆ บนวินโดวส์มาก่อน อีกทั้งยังสามารถกำหนดข้อมูลที่เป็นแบบ Pointer และกำหนดหน่วยความจำ ณ ตำแหน่งต่าง ๆ ได้อีกด้วย ด้วยรูปแบบการใช้ภาษาและการทำงานของ Delphi ทำให้มีผู้กล่าวไว้ว่า Delphi เป็นเครื่องมือในการพัฒนา Application บนพื้นฐานของ Object-Oriented ที่มีความสมบูรณ์มาก โดยสามารถใช้งานคุณสมบัติในการ Encapsulation, inheritance และ Polymorphism ได้ทั้งหมด และยังขยายขีดความสามารถอื่น ๆ ออกไปอีก เช่น ในเรื่องของ การกำหนด Properties ของออปเจ็กต์, Privacy, การตรวจสอบข้อมูลในช่วงเวลา Run-time ได้, Exception handling, การอ้างอิงถึงคลาสต่าง ๆ เป็นต้น

นอกจากนี้ Delphi ยังสนับสนุนรูปแบบในการเขียนโปรแกรมเพื่อการควบคุมในระบบต่าง ๆ ได้อย่างค่อนข้างเสรีอีกด้วย ยกตัวอย่างเช่น

- การกำหนด Subclass ของการควบคุมในวินโดวส์
- การ Overide เมสเสจของวินโดวส์
- การควบคุมเมสเสจบางอย่างของวินโดวส์
- ส่วนของโพรซีเดอร์สามารถเขียนด้วยภาษาแอสเซมบลีได้ด้วย

สภาพแวดล้อมที่กำหนดได้

โดยแบบฉบับของบอร์แลนด์ในการพัฒนาเครื่องมือสำหรับการเขียนโปรแกรมที่ได้มีการปรับปรุงมาโดยตลอด ซึ่งจากรูปแบบเดิมก็เป็นแบบ Command line Compiler จนกระทั่งได้รวมเอา Editor และ Compiler เข้าด้วยกัน และปรับปรุงมาเป็นสภาพการเขียนโปรแกรมบนเครื่องมือแบบ IDE (Integreted Development Environment) ที่ได้มีการรวมเอาวินโดว์ต่าง ๆ เข้าด้วยกัน มีการตรวจสอบการทำงานของโปรแกรมเป็นขั้น ๆ รวมถึงการตรวจดูค่าที่กำหนดไว้ได้อีกด้วย



รูปที่ 9 แสดงการกำหนด Environment ต่าง ๆ

เครื่องมือในการพัฒนาโปรแกรมใน Delphi ได้ออกแบบมาให้อย่างสมบูรณ์ พร้อมทั้งสามารถให้ผู้ใช้กำหนดรูปแบบบางอย่างได้เองอีกด้วย เช่น การกำหนดรูปแบบการควบคุมจากแป้นพิมพ์ใหม่ให้เป็นแบบมาตรฐานในวินโดวส์, BRIEF, Epsilon/Emancs หรือแบบ DOS IDE ก็ขึ้นอยู่กับว่าใครถนัดแบบไหน รวมถึงการจัดปุ่มบน Speedbar, Component Palette, ค่า Default ต่าง ๆ และสำหรับใครที่ยังใช้เครื่อง Pentium รุ่นแถม Bug หากกลัวการคำนวณที่ผิดพลาดก็กำหนดให้โปรแกรมป้องกันไว้ก่อนก็ยังได้ (เลือกที่ Project Options -> Complier -> Pentium-safe FDIV)

ลองของจริง

เท่าที่เคยใช้งาน VB มา จุดหนึ่งที่ผู้เขียนรู้สึกอึดอัดกับการพัฒนาโปรแกรมบน VB ก็คือ จำนวนของโปรแกรมที่ประกอบด้วยโปรแกรมย่อย ๆ จำนวนมากและความเร็วในการทำงาน และเมื่อได้มีโอกาสจับ Delphi เข้าจริง ๆ ก็สามารถแก้ปัญหาในจุดต่าง ๆ เหล่านี้ได้หมดไป กล่าวคือ ไฟล์ที่สร้างมาจาก Delphi จะเป็นไฟล์ที่สามารถนำไปใช้งานได้ทันที ไม่ต้องมีการติดตั้งอะไรเพิ่มเติม และความเร็วในการทำงานก็เป็นที่น่าพอใจมาก เมื่อเทียบกับความรู้สึกที่เห็นกับคำโฆษณาที่บอกว่าเร็วกว่า 10-20 เท่า

ลิสติงที่ 1

```
VB - Fill Database

Private Sub Fill_Button_Click()
Dim n, MaxRecord As Integer
      MaxRecord = Text1.Text
      tm1 = Timer
      For n = 1 To MaxRecord
            Data1.Recordset.AddNew
            Data1.Recordset("N_FIRST") = "Sanya " + Str(n)
            Data1.Recordset("N_LAST") = "Kasetvetin " + Str(n)
            Data1.Recordset("AGE") = n
            Data1.Recordset("PHONE") = Str(n)
            Data1.Recordset.Update
      Next n
      Data1.Recordset.MoveLast
      label1.Caption = Format(Timer - tm1, "####0.000")
End Sub


VB - Read Database

Private Sub Read_Button_Click()
Dim n, MaxRecord As Integer
Dim NF, NL, PH As String
Dim AG As Integer
      MaxRecord = Text1.Text
      tm1 = Timer
       Data1.Recordset.MoveFirst
      For n = 1 To MaxRecord
            NF = Data1.Recordset("N_FIRST")
            NL = Data1.Recordset("N_LAST")
            AG = Data1.Recordset("AGE")
            PH = Data1.Recordset("PHONE")
            Data1.Recordset.MoveNext
      Next n
      Label2.Caption = Format(Timer - tm1, "####0.000")
End Sub


VB - Sorting

Private Sub Sort_Button_Click()
Const Max As Integer = 6000
Dim SortArray(Max) As Integer
Dim i, j As Integer
Dim temp As Integer
```

→

ตารางที่ 1

	Fill		Read		Sort	
#item	VB	Delphi	VB	Delphi	VB	Delphi
100	1.27	0.16	0.5	0.00	0.11	0.0
500	5.16	0.66	1.93	0.33	1.93	0.11
1000	10.16	1.21	3.84	0.71	7.53	0.49
2000	20.65	2.52	7.63	1.37	29.55	1.87
3000	31.69	3.79	11.54	2.90	66.31	4.12
4000	43.11	5.60	15.38	3.02	117.48	7.42
5000	54.98	6.26	19.17	3.50	183.18	11.53

หมายเหตุ
- หน่วยที่ใช้วัดเป็นวินาที
- ค่าที่ได้เป็นค่าเฉลี่ย และอาจเปลี่ยนแปลงตามระบบของเครื่องที่ใช้

ลิสติงที่ 1 (ต่อ)

```
      MaxArray = Text1.Text
      ' fill up array
      For i = 1 To MaxArray
            SortArray(i) = i
      Next i
      ' Begin sort
      tm1 = Timer
      For i = 1 To MaxArray - 1
            For j = 1 To MaxArray - i
            If SortArray(j) < SortArray(j + 1) Then
                temp = SortArray(j)
                SortArray(j) = SortArray(j + 1)
                SortArray(j + 1) = temp
            End If
        Next j
    Next i
    Label3.Caption = Format(Timer - tm1, "####0.000")
End Sub
```

Delphi - Fill Database

```
Procedure TForm1.Fill_ButtonClick(Sender: TObject);
Var n : Integer;
      s,ep : String;
      tm1 : Real;
      hh,mm,ss,ms : Word;
      MaxRecord,err_code : Integer;
Begin
        Val(edit1.text,MaxRecord,err_code);
      tm1 := time;
      For n:=1 To MaxRecord Do
      With table1 Do
         Begin
           Str(n,s);
           Append;
             FieldByName('N_FIRST').AsString := 'Sanya '+s;
           FieldByName('N_LAST').AsString := 'Kasetvetin '+s;
            FieldByName('AGE').AsInteger := n;
            FieldByName('PHONE').AsString := s;
            Post;
        End;
      DecodeTime((time-tm1),hh,mm,ss,ms);
      Label1.Caption := IntToStr(mm)+':'+IntToStr(ss)+':'+IntToStr(ms);
End;
```

Delphi - Read Database

```
Procedure TForm1.Read_ButtonClick(Sender: TObject);
Var n, AG : Integer;
      NF,NL,PH : String;
      s,ep : string;
      tm1 : Real;
      hh,mm,ss,ms : Word;
      MaxRecord,err_code : Integer;
Begin
       Val(edit1.text,MaxRecord,err_code);
      tm1 := time;
      Table1.First;
      For n:=1 To MaxRecord Do
      With Table1 Do
       Begin
            NF := FieldByName('N_FIRST').AsString;
            NL := FieldByName('N_LAST').AsString;
            AG := FieldByName('AGE').AsInteger;
            PH := FieldByName('PHONE').AsString;
       Table1.Next;
     End;
     DecodeTime((time-tm1),hh,mm,ss,ms);
     Label2.Caption := IntToStr(mm)+':'+IntToStr(ss)+':'+IntToStr(ms);
End;
```

→

ลิสติงที่ 1 (ต่อ)

```
Delphi - Sorting

Procedure TForm1.Sort_ButtonClick(Sender: TObject);
Const Max = 6000;
Var i,j : Integer;
      temp : Integer;
      SortArray : Array[1..Max] of Integer;
      MaxArray,err_code : Integer;
      hh,mm,ss,ms : word;
      tm1 : Real;
Begin
      Val(edit1.text,MaxArray,err_code);
      { Fill up array }
      For i := 1 To MaxArray do
         SortArray[i] := i;
      { Begin sort }
      tm1 := time;
      For i := 1 To MaxArray - 1 Do Begin
         For j := 1 To MaxArray - i Do Begin
            If SortArray[j] < SortArray[j+1] Then Begin
               temp := SortArray[j];
               SortArray[j] := sortArray[j+1];
               SortArray[j+1] := temp;
            End;
        End;
     End;
       DecodeTime((time-tm1),hh,mm,ss,ms);
       label13.caption := IntToStr(mm)+':'+IntToStr(ss)+':'+IntToStr(ms);
End;

End.
```

ในตอนแรกเองก็ไม่อยากจะเชื่อต่อคำโฆษณามากนักก็เลยต้องมานั่งเขียนโปรแกรมตรวจสอบดูเอง สำหรับโปรแกรมที่ตรวจสอบดูนี้อาจจะไม่เป็นการตรวจสอบทั้งหมดทุกเรื่อง แต่ก็ได้เขียนเพื่อตรวจสอบใน 3 เรื่องคือ การเขียนข้อมูลลงไฟล์ฐานข้อมูล การอ่านข้อมูลจากไฟล์ฐานข้อมูล และการเรียงลำดับข้อมูล

ขั้นตอนการทดสอบ

- การทดสอบนี้กระทำบนเครื่อง 486DX2-66 หน่วยความจำ 8 เมกะไบต์ ไม่มีการติดตั้ง Cache ใด ๆ และทำงานบนสภาพแวดล้อมแบบเดียวกัน

- ไฟล์ฐานข้อมูลที่ใช้ในการทดสอบเป็นไฟล์ที่มาจากโปรแกรม dBase ประกอบด้วย 4 ฟิลด์

- ทดสอบโดยใช้โปรแกรม Delphi รุ่น Desktop และโปรแกรม Visual BASIC 4.0 (16 บิต)

- เขียนโปรแกรมเพื่อเขียนข้อมูลลงไฟล์ อ่านข้อมูลจากไฟล์ใส่ตัวแปร เรียงลำดับข้อมูลตัวเลขด้วยโปรแกรม Delphi และ Visual BASIC ทำการคอมพายล์เป็น EXE แล้ว Run ทดสอบ ได้ผลดังตารางที่ 1

- ส่วนของโปรแกรมที่ใช้ตรวจสอบแสดงไว้ดังลิสติงที่ 1

การจับเวลาได้เขียนคำสั่งในการเรียกใช้นาฬิกาภายในเครื่องมาเป็นเครื่องวัด และให้แสดงผลออกมาใน Label ที่กำหนดไว้ จุดหนึ่งซึ่งอาจจะเห็นว่า Delphi ได้เปรียบเหนือกว่า VB ก็คือ การอ่านและเขียนแฟ้มฐานข้อมูล เนื่องจากได้ใช้แฟ้ม DBF เป็นแฟ้มในการตรวจสอบ ซึ่งในจุดนี้เอง VB จะต้องเรียกใช้ไฟล์ผ่านทาง ODBC ในตอนแรกผู้เขียนได้เขียนโดยทดลองอ่านจากไฟล์ MDB แต่ก็ประสบกับปัญหาในการอ่านเวอร์ชัน 1.X และ 2.0 ก็เลยต้องเปลี่ยนมาใช้ไฟล์ DBF เอง แต่อย่างไรก็ตาม ในการเรียงลำดับก็คงจะเป็นเครื่องพิสูจน์ได้ถึงความเร็วในการทำงานได้ เนื่องจากเป็นการทำงานของโปรแกรมจริง ๆ ไม่มีการเรียกผ่านไดรเวอร์ใด ๆ

ส่งท้าย

บทความทั้งหมดนี้เป็นข้อมูลและแนวความคิด สำหรับท่านที่กำลังตัดสินใจจะเลือกแนวทางการพัฒนาโปรแกรมบนวินโดวส์ ไม่ว่าจะต้องการเขียนเพื่อศึกษาภาษา หรือเขียนเพื่องานธุรกิจ Delphi ก็นับว่าเป็นทางออกที่ดีในการพัฒนาโปรแกรม

สำหรับท่านที่ยังไม่เคยมีพื้นความรู้ในการเขียนโปรแกรมด้วยภาษาปาสคาลมาก่อน ก็สามารถจะศึกษาจาก Delphi ได้โดยตรง โดยใช้ WinCrt (อ่านจากตัวอย่างหรือคู่มือในโปรแกรมก็ได้) ซึ่งเป็นโครงสร้างภาษาที่เหมือนกับในดอส

หากท่านมีความคิดเห็นเพิ่มเติม หรือมีข้อเสนอแนะ สามารถติดต่อกับผู้เขียนได้โดยตรง ผ่านทาง E-mail มาที่

sanya@cs1.cmma.ac.th